

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 08237532 A

(43) Date of publication of application: 13.09.96

(51) Int. Cl

H04N 5/232

(21) Application number: 07037913

(71) Applicant: CANON INC

(22) Date of filing: 27.02.95

(72) Inventor: OKAZAKI HIROSHI
SATO HIROAKI

(54) REMOTE CONTROL SYSTEM FOR CAMERA

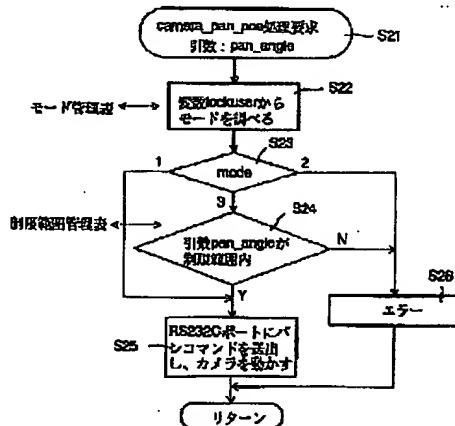
(57) Abstract:

PURPOSE: To set a suitable photographing enable range by providing a mode for permitting all camera operations, mode for inhibiting all the camera operations and mode for permitting the camera operation within a limited range in a control right managing means.

CONSTITUTION: The complete permission mode is defined as mode 1, the full inhibition mode is defined as mode 2, and the limited permission mode is defined as a mode 3. A camera managing server stores and manages a mode managing table showing an access limit mode and the limited range showing the permitted range in the limited permission mode. Corresponding to a camera panning request for designating a panning angle, the access limit mode of a request source is investigated from a variable Lockusev and in the case of mode 3, it is confirmed by the limited range managing table that the designated panning angle is within the limited range. When it is within the limited range, a panning command is sent out from an RD232C and a camera apparatus is moved at the designated panning angle. In the case of mode 1, the camera is moved at the designated panning angle by passing S24. In the case of

mode 2, it is reported to the request source that remote control is not permitted.

COPYRIGHT: (C)1996,JPO



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-237532

(43)公開日 平成8年(1996)9月13日

(51)Int.Cl.^o

H 04 N 5/232

識別記号

府内整理番号

F I

H 04 N 5/232

技術表示箇所

B

審査請求 未請求 請求項の数5 OL (全10頁)

(21)出願番号 特願平7-37913

(22)出願日 平成7年(1995)2月27日

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 岡崎 洋

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(72)発明者 佐藤 宏明

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

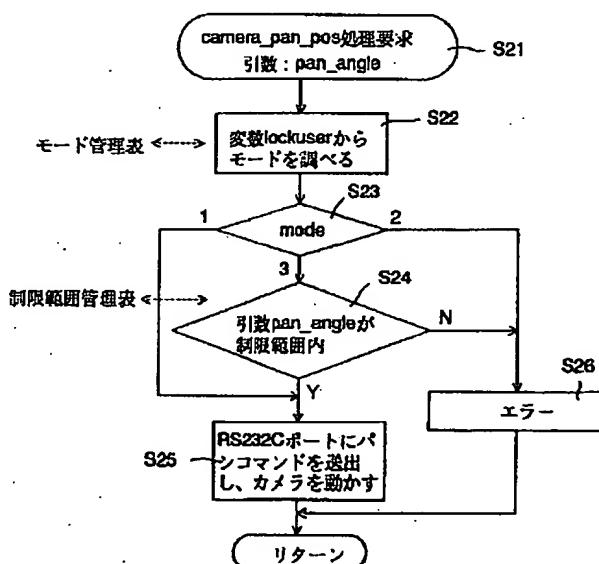
(74)代理人 弁理士 田中 常雄

(54)【発明の名称】 カメラの遠隔制御システム

(57)【要約】

【目的】 制限された範囲のカメラ操作をユーザ毎に設定できるようにする。

【構成】 全てのカメラ操作を許容する完全許可モード(モード1)と、一切のカメラ操作を許さない全面禁止モード(モード2)と、これらの中間として制限された範囲内で制御できる限定許可モードの3種類のアクセス制限モード(モード3)を設け、各ユーザ毎に設定する。カメラ操作要求に対して、ユーザのモードを調べる(S23)。モード3では、制限範囲管理表により指定された範囲でカメラ装置を駆動する(S24, 25)。モード1では、指示された状態にカメラ装置を駆動する(S25)。モード2では、遠隔操作が許可されていないことを要求元に通知して、終了する(S26)。



【特許請求の範囲】

【請求項1】 ネットワークを介して遠隔制御自在なカメラ装置の遠隔制御方式であって、当該カメラ装置を同時に一人だけが制御できるように制御権を管理する制御権管理手段を設け、当該カメラ装置を制御するユーザごとに、全てのカメラ操作を許容する第1のモードと、全てのカメラ装置を禁止する第2のモードと、制限範囲のカメラ操作を許容する第3のモードのいずれかを設定できるようにしたことを特徴とするカメラの遠隔制御システム。

【請求項2】 第3のモードにおける制限範囲が複数種類、設定自在である請求項1に記載のカメラの遠隔制御システム。

【請求項3】 更に、前記カメラ装置を制御する1以上の制御装置をシステムに登録し、及び当該システムから削除する登録手段と、前記カメラ装置への当該制御手段の接続を設定する接続設定手段と、当該システムに登録された前記制御装置の前記カメラ装置に対する接続状態を管理する接続状態管理手段とを有する請求項1又は2に記載のカメラの遠隔制御システム。

【請求項4】 更に、前記制御装置の前記カメラ装置に対する接続状態を表示する表示手段を有する請求項3に記載のカメラの遠隔制御システム。

【請求項5】 更に、前記カメラ装置に対する前記制御手段の接続の可否を管理する接続可否管理手段を具備する請求項3又は4に記載のカメラの遠隔制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、カメラの遠隔制御システムに関し、より具体的は、例えばビデオ会議用カメラを遠隔操作するカメラの遠隔制御システムに関する。

【0002】

【従来の技術】 パン、チルト及びズームなどを遠隔操作できるカメラは周知であり、例えば、遠隔監視システムやTV会議システムで利用されている。但し、TV会議システムは、一般には、会議室等に設置される専用システムであり、会議に参加できる端末は限定されているのが普通である。即ち、会議メンバでない端末が会議途中に新規に参加することは予定されていない。予め会議メンバになっておくことが必要であり、そのための設定操作がシステム管理者により行なわれる。

【0003】

【発明が解決しようとする課題】 しかし、近年、高速なコンピュータ・ネットワークが急速に普及し、ワークステーション又はパーソナル・コンピュータの機能向上と低価格化が進み、その結果、パーソナル・コンピュータ等を使って多数の不特定の人々が参加できるビデオ会議システムが実現可能になってきた。

【0004】 このような、不特定者の参加が予想されるビデオ会議システムでは、多数のユーザが必要に応じて

ビデオ会議に参加し、退席することになり、その結果、ユーザの会議への参加及び離脱（ネットワークへの端末機器の接続及び分離）、会議に参加しているユーザの、相互の端末機器に対してのアクセス許可の設定、並びに、ユーザ間でのアクセス状態の告知等のシステム管理が、重要になってくる。

【0005】 特に、カメラ制御に関して、その許可と解放をダイナミックに管理できる必要がある。更には、会議室等に置かれる専用端末以外には、次のような配慮も必要になる。即ち、一般的のコンピュータは、その主たる用途に合わせて配置されるので、その場所にビデオ会議のためのビデオ・カメラを設置すると、他人には見られたくない映像が背景などとして相手に伝送されてしまう。ビデオ会議のためだけに背景にカーテンなどを設けるのは不自然であり、面倒である。自端末なビデオ・カメラを相手端末から制御できないようにしてもよいが、それでは、スムーズな会話が困難になったり、人物及び資料などを適時に適切な大きさ及び適切な位置で撮影し、伝送できるという利点が損われてしまう。

【0006】 そこで、本発明は、このような問題点を解決したカメラの遠隔制御システムを提示する事を目的とする。

【0007】

【課題を解決するための手段】 本発明では、ネットワークを介して遠隔制御自在なカメラ装置を同時に一人だけが制御できるように制御権を管理する制御権管理手段を設け、当該カメラ装置を制御するユーザごとに、全てのカメラ操作を許容する第1のモードと、全てのカメラ装置を禁止する第2のモードと、制限範囲のカメラ操作を許容する第3のモードのいずれかを設定できるようにした。

【0008】

【作用】 上記手段により、ユーザ毎に、カメラ操作を完全に許すと及び完全に禁止する以外に、制限された範囲でのカメラ操作を許すことができるようになる。これにより、撮影されたくない部分がある場合に、ユーザ毎にその部分の撮影禁止を設定できるようになる。また、撮影されたくない部分がユーザ毎に異なる場合にも、その要求を満たす事が出来る。

【0009】

【実施例】 以下、図面を参照して本発明の実施例を詳細に説明する。

【0010】 図1は、本発明の一実施例の概略構成ブロック図である。図1に示す実施例では、3台のワークステーション10A, 10B, 10Cがネットワーク40に接続している。各ワークステーション10A, 10B, 10Cは、CPU12A, 12B, 12C, ROM, RAM及びハードディスク装置（外部記憶装置）を有する記憶部14A, 14B, 14C、入出力部としてのI/Oポート16A, 16B, 16C; 18A, 18

B, 18C及びビデオ・ボード20A, 20B, 20Cを具備し、これらはバス22A, 22B, 22Cに接続している。バス22A, 22B, 22Cにはまた、ポインティング・デバイスとしてのマウス24A, 24B, 24Cとキーボード26A, 26B, 26Cが接続する。

【0011】ワークステーション10A, 10Cにはそれぞれ、そのI/Oポート16A, 16Cを介して、パン、チルト及びズームを外部制御できるカメラ装置28A, 28Cが接続する。各カメラ装置28A, 28Cは、ズーム・レンズを具備するカメラ部30A, 30Cと、カメラ部30A, 30Cをパン及びチルトする雲台部32A, 32Cとからなる。

【0012】各ワークステーション10A, 10B, 10Cは、そのビデオ・ボード20A, 20B, 20Cに、モニタ・ディスプレイ装置（以下、モニタと略す。）34A, 34B, 34Cが接続する。ビデオ・ボード20A, 20Cは、カメラ装置28A, 28Cからの映像及び／又はネットワーク40を介して受信した映像をモニタ34A, 34Cに表示する。また、ビデオ・ボード20Bは、ネットワーク40を介して受信した映像をモニタ34A, 34Cに表示する。勿論、モニタ34A, 34B, 34Cには、以下に説明するカメラ制御パネルを含むその他の情報も表示することができる。

【0013】ワークステーション10Bにはカメラ装置28A, 28Cと同様のカメラ装置が接続されていないので、画像を他のワークステーション10A, 10Cに伝送できない。

【0014】いまでもないが、CPU12A, 12B, 12Cは、各ワークステーション10A, 10B, 10C及びネットワーク40を介した通信を制御し、その制御プログラムは、記憶部14A, 14B, 14CのROM又はハードディスク装置に記憶されており、RAMには各種のデータが記憶される。I/Oポート16A, 16Cは、CPU12A, 12B, 12Cからのカメラ制御信号をカメラ装置28A, 28Cに転送し、カメラ装置28A, 28Cからのカメラ状態信号をCPU12A, 12Cに転送する。I/Oポート16A, 16Cは、本実施例では、RS232C規格でカメラ装置28A, 28Cに接続する。

【0015】各ワークステーション10A, 10B, 10Cは、そのI/Oポート18A, 18B, 18Cを介してネットワーク40に接続する。

【0016】このような基本的なハードウェア構成により、ビデオ会議を実施できる。なお、本実施例では、汎用性及び処理能力を考慮してワークステーションを利用しているが、近年の処理能力の向上を考慮すれば、パソコン・コンピュータでも実現可能であることは明らかである。また、ワークステーションの代わりに専用の制御装置を用いても構わない。

【0017】図2は、本実施例を実行するためのソフトウェア構成を示す。図2では、カメラ装置が直接接続する端末にインストールするソフトウェア（サーバ）と、そのカメラ装置を遠隔操作しようとする端末にインストールするソフトウェア（クライアント）を分けて、図示してある。勿論、通常の利用形態では、1つの端末にどちらのソフトウェアもインストールされることになる。

【0018】図2では、ワークステーション50, 52, 54, 56がネットワーク57に接続し、ワークステーション50には、カメラ装置28A又は同28Cであるカメラ装置58が接続し、ワークステーション52, 654には、それぞれモニタ60, 62が接続しているとする。ワークステーション56は、詳細は後述するが、ビデオ会議で利用可能なカメラ装置のアクセスを管理する。

【0019】ワークステーション50には、カメラ装置58によって得られた画像情報をネットワーク57を介して他のワークステーションに送信するビデオ会議ソフトウェア64と、他のワークステーション52, 54からのカメラ制御信号に応じてカメラ装置58を制御するカメラ制御サーバ（ソフトウェア）66が格納されている。ワークステーション56にはカメラ管理サーバ（ソフトウェア）68がインストールされており、カメラ制御サーバ66は、カメラ装置58の起動時に、カメラ管理サーバ68にカメラ装置58がビデオ会議システム（ネットワーク57）に接続されたこと、即ち現実に利用可能ことの登録を要求し、カメラ管理サーバ68はその事実を記憶する。

【0020】ワークステーション52, 54には、ワークステーション50からの画像をモニタ60, 62に表示するビデオ会議ソフトウェア70と、カメラ装置（ここでは、カメラ装置58）を遠隔制御するためのカメラ制御パネルを表示し、その操作を処理するカメラ制御クライアント（ソフトウェア）72とが格納されている。ワークステーション52, 54のユーザは、カメラ制御クライアント72により、カメラ管理サーバ68で管理されている種々の情報を参照し、記入し、訂正することができる。詳細は後述する。なお、カメラ管理サーバ68により他のワークステーションのカメラ装置（ここではカメラ装置58）との接続（撮影画像の受信と遠隔制御）が許可されると、カメラ制御クライアント72は、カメラ管理サーバ68を介すことなく、カメラ制御信号をカメラ制御サーバ66に送信できる。

【0021】図2では、カメラ管理サーバ68が専用のワークステーションに搭載されているが、ビデオ会議のネットワーク57に参加する何れかのワークステーション（図2では、ワークステーション50, 52, 54）に搭載されていてもよいことは勿論である。ビデオ会議への参加を予定する全端末に予め装備しておき、最初に

ビデオ会議を開始する2つの端末の何れかでカメラ管理サーバ68が起動すればよい。

【0022】通常の使用法では、各ワークステーションにはモニタとカメラ装置の両方が接続しているのが普通であることから、ビデオ会議ソフトウェアは、ビデオ会議のための単一のプログラム又は一群のプログラム・モジュールからなる。その他のソフトウェアも、一般に、単一のプログラム又は一群のプログラム・モジュールからなる。図1に示すワークステーション10A, 10Cは、図2に示すワークステーション50と同52又は54からなり、ワークステーション10Bは、ワークステーション52又は同54からなる。図2に示すワークステーション56は、図1では、ワークステーション10A, 10B又は10Cである。

【0023】次に、図2に示す構成でのカメラ制御とユーザ管理を説明する。

【0024】まず、ワークステーション52又は同54から他のワークステーション50に接続されるカメラ装置58を遠隔制御可能とするための手順を説明する。ワークステーション50のカメラ制御サーバ66はRS232Cを介してカメラ装置58にカメラ制御信号を送り、カメラ装置58を制御できる。本実施例では、ワークステーション52, 54は、ワークステーション50のカメラ制御サーバ66に含まれるカメラ制御関数（パン、チルト及びズーム等のカメラ制御に関する関数）をRPC（Remote Procedure Call）により遠隔起動できる。これにより、ワークステーション52, 54は間接的に、カメラ装置58を遠隔操作できることになる。起動されたカメラ制御関数は、対応する内容のコマンド列をRS232Cポートからカメラ装置58に出力する。このコマンド列のうちで、パン及びチルト等の雲台の動作に関するものは、雲台の制御部に送信され、ズーム等のカメラ部の動作に関するものが、カメラ部の制御部に送信される。

【0025】本実施例では、カメラ制御クライアント72からRPCを用いて遠隔地のカメラ制御サーバ66を呼び出す関数として、図3に示すような関数を用意する。以下では、図に示した関数又はコマンド等を[]でくくって記載することにする。本明細書では、C言語で各関数を記述しているが、プログラム言語が異なれば、異なる表記になることは明らかである。

【0026】図3において、[struct camera_in {~} camera_in]は、カメラ装置58のために、カメラ制御クライアント72からカメラ制御サーバ66に引き渡すパラメータ・セットを設定するデータ宣言であり、ここでは、チルト、パン及びズームのパラメータを設定できるようになっている。また、[struct func_out {~} func_out]は、関数の出力値を設定するデータ宣言であり、例えば、通常通り、関数の処理結果に問題がな

い場合は0を返し、問題がある場合には1を返すようとする。[camera_open]は、カメラ制御サーバ66との通信に必要なクライアント・ハンドラ[CLIENT *c1]を作成する関数であり、[camera_close]は、これを消去する関数である。クライアント・ハンドラ[CLIENT *c1]は、制御しようとするカメラ装置を特定する。

【0027】1台のカメラ装置を同時に1人のユーザだけが遠隔制御できるように、アクセス権を一時的に制限する必要がある。そのため、図4に示すような関数[lock_1]と[unlock_1]を設けた。関数[lock_1]内では、ロック状態を示すサーバ内変数(lock_status)を参照し、既にロック状態であればロック失敗（即ち、エラー）を返す。非ロック状態であればロック状態に変えて、ホスト名とユーザ名を登録し、ロック成功を返す。関数unlock_1内では、既にロック状態になっていて、先に登録したホスト名とユーザ名が引数で指定されたものと一致する場合に、ロック状態を非ロック状態に変え、アンロック成功を返す。その他の場合にはアンロック失敗としてエラーを返す。

【0028】カメラ制御クライアント72は、最初に[camera_open]を呼ぶことによりカメラ制御サーバ66との通信路を確保し、その後、関数[lock_1]により他のユーザによる制御を禁止し、[camera_pan_pos_1]等の関数によりカメラ装置58の制御をカメラ制御サーバ66に命令する。カメラ制御を終了したら、関数[unlock_1]で自己のアクセス権を放棄し、[camera_close]によりカメラ装置58への通信路を閉じる。ロック関数は、カメラ装置の具体的な操作の関数の実行前後に挿入するようにしてよい。その場合には、1台のカメラ装置の制御権を1人のユーザが長期に専有してしまうことを防止できる。

【0029】次に、カメラ管理サーバ68によるユーザの管理を説明する。カメラ管理サーバ68は、図5に示すような構造のカメラ状態リスト80により、ユーザを管理する。カメラ状態リスト80には、ネットワーク57に接続されているカメラ装置58等を示す名称、並びに、その使用状況（空きか又は他のユーザがアクセス中か等）、設置位置(x, y, z)及び方位（パン角とチルト角）が格納されている。カメラ制御サーバ（例えば、66）が、1台のカメラ装置（例えば、58）しか制御しない場合には、カメラ制御サーバが稼働するワークステーションのユーザ名を、カメラ装置の名称としてしてもよいが、2台以上のカメラ装置を制御している場合には、個々のカメラ装置に別の名称を付けて区別することになる。

【0030】図5に示すカメラ状態リスト80では、4台のカメラ装置が例示され、それぞれの名称をhost

1, host 2, host 3及びhost 4としている。カメラ状態リスト80はまた、host 2のカメラ装置をhost 3のユーザが制御し、host 4のカメラ装置をhost 1のユーザが制御している状態にあることを示している。host 1とhost 3は、いずれのユーザにも操作されていない状態にある。

【0031】ユーザの登録及び削除の処理を説明する。カメラ制御サーバ66は、その起動時に、起動したユーザ名（利用可能なカメラ装置の名前）をカメラ管理サーバ68に通知してカメラ状態リスト80に登録させ、カメラ制御サーバ66の停止時に、カメラ管理サーバ68に通知して、カメラ状態リスト80からユーザ名を削除させる。これらは、RPCを用いて、図6に示す関数により実現される。図6において、[struct host_name {~} host_name]は、ホスト名を設定するデータ型宣言であり、MAXNAMEは、設定されるホスト名の最大文字数を規定する。[append_camera_list_1]及び、[delete_host_list_1]は、それぞれ、カメラ制御サーバ66の起動時及び停止時に起動され、カメラ管理サーバ68内で実行される関数である。

【0032】カメラ制御サーバ66は、その起動時に、[append_camera_list_1]を起動する。これにより、図7に示すように、カメラ管理サーバ68は、カメラ状態リスト80に利用可能になったカメラ装置のホスト名を登録する。ホスト名は、図6からわかるように、host_name型で構成され、具体的には、[host_name->name]で示される。これ以降、登録されたホスト名で特定されるカメラ装置（図2では、58）を、担当するカメラ制御サーバ（図2では、66）を介して他のワークステーションから制御可能になる。

【0033】カメラ制御サーバ66はまた、その停止時に、[delete_host_list_1]を起動する。これにより、図8に示すように、カメラ管理サーバ68は、カメラ制御サーバ66により制御されていて、以後利用できなくなるカメラ装置を、カメラ状態リスト80から削除する。

【0034】次に、カメラ管理サーバ68によるアクセス管理を説明する。カメラ管理サーバ68によるアクセス管理のために、カメラ管理サーバ68において実行される図9に示す関数[access_begin_1]及び[access_end_1]を設けた。[access_begin_1]はアクセスの開始処理を実行し、[access_end_1]はアクセスの終了処理を実行する。[struct access_in {~} access_in]は、アクセスの対象をパラメータ列の宣言であり、カメラ制御クライアント72からカメラ管理サーバ68に引き渡される。[target_name]は、アクセスする相手を特定し、[u

ser_name]はユーザ（自分）を特定する。

【0035】カメラ管理サーバ68は、図10に示すアクセス許可リスト82を備える。アクセス許可リスト82は、先に、カメラ状態リスト80に登録された個々のカメラ装置ごとに、ビデオ会議に参加している又は参加する可能性のあるユーザのアクセス（即ち、遠隔制御と画像の受信）の権利の有無を保存するものである。図10に示す例では、host 2で示されるカメラ装置に対して、host 1及びhost 4で示されるユーザがアクセス許可されて、host 3で示されるユーザはアクセスが禁止されている。なお、host 2のユーザは、host 2で示されるカメラ装置をカメラ管理サーバ68を介すことなく制御できるので、このアクセス許可リスト82の対象外である。

【0036】カメラ制御クライアント72からのアクセス要求は、図11に示すフローチャートに従って処理される。何れかのカメラ制御クライアント72は、何れかのカメラ装置（ここでは、カメラ装置58）を利用（画像受信とカメラ制御）したい場合、対象のカメラ装置名を特定したアクセス要求をカメラ管理サーバ68に送信する。カメラ管理サーバ68はまず、アクセス要求されたカメラ装置名を[target_name]により解読する（S1）。

【0037】次に、カメラ状態リスト80を参照して、指定されたカメラ装置が起動しているか否かを確認する（S2）。起動していないければ、指定されたカメラ装置が未だ起動していないことを要求元に通知する（S7）。

【0038】指定されたカメラ装置が起動していれば（S2）、[user_name]を解読し、この名前で示されるユーザがアクセスを要求するカメラ装置にアクセスする権利を有するかどうかをアクセス許可リスト82を参照して確認する（S3）。アクセス許可がなければ、許可が無い事を要求元に通知する（S7）。

【0039】アクセスが許可されていれば（S3）、指定されたカメラ装置が他のユーザに使用されているかどうかを、カメラ状態リスト80を参照して確認する（S4）。使用されていれば、他のユーザが使用していることを要求元に通知する（S7）。

【0040】他のユーザが使用していなければ（S4）、カメラ状態リスト80にユーザ名（ホスト名）を登録し（S5）、アクセス要求成功を要求元に通知する（S6）。

【0041】なお、カメラ制御クライアント72は、アクセス操作を行なう前後に必ず、[access_begin_1]及び[access_end_1]を実行する。

【0042】本実施例では、各ユーザ又は特に許可されたユーザは、自身で、アクセス許可リスト82の内容を変更できる。そのために、図12に示す関数[chan

`ge_access_mode_1]` をカメラ管理サーバ 68 に設けてある。

【0043】カメラ制御クライアント 72 は、`[struct change_in {~} change_in]` によって、アクセス許可を変更するホスト名（カメラ装置）を`[target_name]` に設定すると共に、変更後のアクセス権を`[access_mode]` に設定して、`[change_access_mode_1]` を呼ぶ。カメラ管理サーバ 68 は、まずUNIXオペレーティング・システムのライブラリ関数である`[gethost]` 命令を実行し、どのユーザからの変更要求かを確認する。次に、アクセス許可リスト 82 のうち、変更要求を出したユーザの項目で、`[target_name]` で示されるカメラ装置のアクセス権を、`[access_mode]` で示されるモードに変更する。`[access_mode]` の内容には、アクセス許可と、アクセス禁止がある。アクセス許可の場合、図 10 に示すように、アクセス許可リスト 82 上で「OK」と記入又は表示され、アクセス禁止の場合、「NO」と記入又は表示される。

【0044】カメラ制御クライアント 72 は、以上の各種の操作を容易に行なえるように、図 13 に示すようなカメラ制御パネルをモニタ画面に表示し、GUI (Graphical User Interface) による操作を実現している。ワークステーション 52, 54 のオペレータは、ビデオ会議に参加しているカメラ装置の登録状況、使用状況及び使用許可状況に関する情報を得ることができ、操作対象のカメラ装置を切り換えることができる。また、アクセス権を変更することもできる。

【0045】図 13 に示すカメラ制御パネルによる操作方法を説明する。このカメラ制御パネルは、ユーザーによるボタン操作に応じて、各ボタンに割り当てられた機能に相当する関数を、RPC により起動する。即ち、ユーザーがカメラ状態リスト更新ボタン 110 を押下すると、これまで述べてきたようにして、カメラ管理サーバ 68 から現在の各カメラ装置のアクセス情報を得ることができる。周知の通り、マウス 138 を操作してモニタ画面上のカーソルを移動させ、所望のボタン上に位置させて、第 1 ボタン 138a をクリックすることで、任意のボタン 110～128 を操作できる。カメラ管理サーバ 68 から得られた情報は、リスト・ウインドウ 136 に表示される。リスト・ウインドウ 136 にはまた、カメラ状態リスト 80 を表示することができ、ユーザーは、リスト・ウインドウ 136 に表示されたカメラ状態リスト 80 から、制御したいカメラ装置をマウス 138 又はキーボードにより選択できる。

【0046】先に説明したアクセス要求処理はアクセス開始ボタン 112 に割り付けられ、アクセス終了処理はアクセス終了ボタン 114 に割り付けられている。ア

セス許可の変更処理は、アクセス許可設定ボタン 116 に割り付けられている。アクセス許可設定ボタン 116 を押下すると、アクセス許可リスト 82 で自身に関係する部分がリスト・ウインドウ 136 に表示され、マウス 138 又はキーボードによりその内容を変更できる。アプリケーション終了ボタン 118 は、すべての操作を終了する時に押下する。

【0047】モニタ画面の画像ウインドウ 134 には、指定したカメラ装置により撮像した画像が表示される。

10 ユーザは、パン操作ボタン 120, 122、チルト操作ボタン 124, 126、ホーム・ポジション復帰ボタン 128、並びに、ズーム（拡大）操作ボタン 130 及びズーム（縮小）操作ボタン 132 を操作することによって、カメラ装置のパン、チルト及びズームを操作できる。この操作に応じて、対象となるカメラ装置がパン、チルト及びズームし、画像ウインドウ 134 に表示される画像も変化する。

【0048】操作ボタン 120, 122, 124, 126 は、パン及びチルトのそれぞれの移動方向に対して設けられている。また、ホームポジション復帰ボタン 128 を設けることにより、制御対象のカメラ装置をホームポジション（基準位置、本実施例ではパン及びチルト方向の回動角度の中央）に簡単に復帰させることができる。ホームポジション復帰ボタン 128 を操作ボタン 120～126 の中央の空きスペースに配置したので、ホームポジションに復帰させるものであることが視覚的にも理解し易くなり、操作性が向上する。なお、ズーム操作については、ズーム操作を指定する 1 つのボタンを設け、そのボタンをマウス 4 の第 1 ボタン 138a 及び

20 第 2 ボタン 138b のどちらでクリックするかで、拡大と縮小を区別するようにしてもよい。このような処理自体はマウス 138 及びキーボードの利用法として周知である。

【0049】実際にビデオ会議を行なう場合を例に、上記実施例の操作を説明する。ビデオ会議の開始に先立って、ビデオ会議ソフトウェア 64, 70、カメラ管理サーバ 68、カメラ制御 66 及びカメラ制御クライアント 72 を立ち上げる。それぞれのソフトウェアの機能を考えると、この順番が望ましいが、カメラ装置 58 の電源 40 オンに連動させて同時に立ち上げてもよい。既に行なわれているビデオ会議に参加する場合、カメラ管理サーバ 68 を新たに立ち上げる必要はないことは明らかである。

【0050】このような状態のもとでビデオ会議に参加する場合、カメラ状態リスト更新ボタン 110 を押下し、リスト・ウインドウ 136 にカメラ状態リスト 80 を表示させ、その中から操作したいカメラ装置を選択し、アクセス開始ボタン 112 を押下する。これにより、前述した`[access_begin_1]` が実行され、前述したアクセス処理及び指定したカメラ装置の

遠隔制御が可能になる。ビデオ会議から抜け出す場合、アクセス終了ボタン 114 を押下する。このとき [access_end_1] が実行される。なお、この段階では、アクセス終了前迄アクセスしていたカメラ装置からの画像は、依然として、ウインドウ 134 に表示されている。すべての操作を終了するときに、アプリケーション終了ボタン 118 を押下する。

【0051】本実施例では更に、個々のユーザに対して、カメラ装置の制御範囲を制限できる。本実施例では、全てのパン、チルト及びズーム操作を許す完全許可モードと、一切のパン、チルト及びズーム操作を許さない全面禁止モードと、これらの中間として制限された範囲内で制御できる限定許可モードの 3 種類のアクセス制限モードを設けた。以下では、完全許可モードをモード 1、全面禁止モードをモード 2、限定許可モードをモード 3 とした。カメラ管理サーバ 68 は、図 14 に示すような、各ユーザについて設定されているアクセス制限モードを示すモード管理表と、限定許可モードでの許可範囲を示す制限範囲管理表を、記憶し、管理する。

【0052】先に説明したロック関数 [lock_1] の実行時に、これらのモード管理表が参照され、モード 2 のユーザに対しては、ロック（アクセス権の確保）が許可されない。図 15 はロックの際のフローチャートを示す。即ち、ユーザ名（遠隔操作しようとする者）とホスト名（操作したいカメラ装置）を示すロック処理要求に対して (S11) 、指定されたカメラ装置がロック状態あるかどうかを調べる (S12) 。既にロック状態にあれば (S12) 、その旨をユーザ名で特定される要求元に通知する (S17) 。

【0053】未だロック状態でなければ (S11) 、モード管理表からそのユーザのアクセス制限モードを調べる (S13) 。モード 2 であれば、制御が禁止されていることを要求元に通知する (S17) 。モード 2 以外であれば、変数 lock_status に 1 をセットしてロック状態にし、変数 lock_user にユーザ名を、変数 lock_host に操作されるカメラ装置名をセットし (S15) 、ロック成功を要求元に通知する (S16) 。

【0054】図 16 は、パン要求に対するフローチャートを示す。パン角を指定するカメラ・パン要求 (S21) に対して、変数 lock_user から要求元のアクセス制限モードを調べ (S22) 、モード 3 のときには (S23) 、制限範囲管理表により、指定されたパン角が制限範囲内にあることを確認する (S24) 。制限範囲内に無ければ、カメラ装置をパンせずに要求元にエラーを通知する (S26) 。制限範囲内にあれば、RS232C からパン・コマンドを送出し、カメラ装置を指定のパン角に動かす (S25) 。モード 1 のときには (S23) 、S24 を迂回して、指定のパン角にカメラ装置を動かす (S25) 。モード 2 のときには、遠隔操作が

許可されていないことを要求元に通知して、終了する (S26) 。

【0055】

【発明の効果】以上 の説明から容易に理解できるように、本発明によれば、システムを停止することなく、多数のユーザが必要に応じてネットワーク上に接続されたカメラ装置を遠隔制御できるようになり、ビデオ会議への参加と離脱が容易になる。更には、カメラの遠隔操作に複数段階のアクセス制限モードを設けたので、ユーザ毎に制御禁止を含めて適切な撮影可能範囲を設定できる。

【図面の簡単な説明】

【図 1】 本発明の一実施例のハードウェアの概略構成ブロック図である。

【図 2】 本実施例のソフトウェアの概略構成ブロック図である。

【図 3】 カメラ制御サーバ 64 を呼び出す関数である。

【図 4】 ロック関数である。

20 【図 5】 カメラ状態リストの一例である。

【図 6】 ユーザの登録及び削除の関数である。

【図 7】 カメラ起動に伴う登録処理のフローチャートである。

【図 8】 カメラ停止に伴う抹消処理のフローチャートである。

【図 9】 アクセス処理の関数である。

【図 10】 アクセス許可リストの一例である。

【図 11】 カメラ・アクセス要求に対する処理のフローチャートである。

30 【図 12】 アクセス許可を設定する関数である。

【図 13】 GUI の一例を示す図である。

【図 14】 ユーザ毎のアクセス制限モードの管理表と、制限範囲の管理表である。

【図 15】 ロック要求に対するフローチャートである。

【図 16】 パン要求に対するフローチャートである。

【符号の説明】

10A, 10B, 10C : ワークステーション

12A, 12B, 12C : CPU

40 14A, 14B, 14C : 記憶部

16A, 16B, 16C, 18A, 18B, 18C : I/O ポート

20A, 20B, 20C : ビデオ・ボード

22A, 22B, 22C : バス

24A, 24B, 24C : マウス

26A, 26B, 26C : キーボード

28A, 28C : カメラ装置

30A, 30C : カメラ部

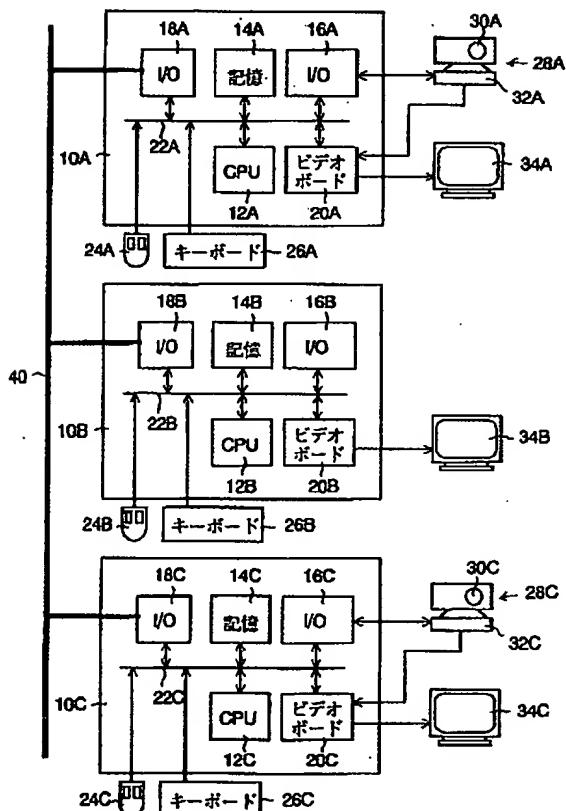
32A, 32C : 雲台部

50 34A, 34B, 34C : モニタ・ディスプレイ

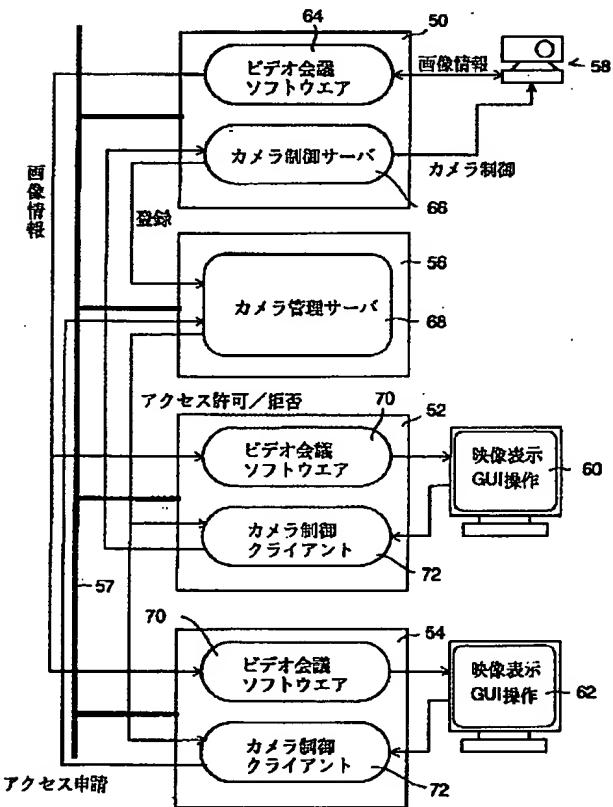
- 40 : ネットワーク
- 50, 52, 54, 56 : ワークステーション
- 57 : ネットワーク
- 58 : カメラ装置
- 60, 62 : モニタ
- 64 : ビデオ会議ソフトウェア
- 66 : カメラ制御サーバ
- 68 : カメラ管理サーバ
- 70 : ビデオ会議ソフトウェア
- 72 : カメラ制御クライアント
- 80 : カメラ状態リスト
- 82 : アクセス許可リスト
- 110 : カメラ状態リスト更新ボタン

112 : アクセス開始ボタン
 114 : アクセス終了ボタン
 116 : アクセス許可設定ボタン
 118 : アプリケーション終了ボタン
 120, 122 : パン操作ボタン
 124, 126 : チルト操作ボタン
 128 : ホーム・ポジション復帰ボタン
 130 : ズーム（拡大）操作ボタン
 132 : ズーム（縮小）操作ボタン
 10. 134 : 画像ウインドウ
 136 : リスト・ウインドウ
 138 : マウス

〔図1〕



[2]



[图4]

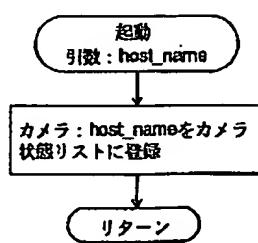
```
typedef struct lock_in {
    char hostname[80];
    char username[80];
} lock_in;
typedef struct lock_out{
    int return_code;
} lock_out;

lock_out *lock_1(lock_in *, CLIENT *cl);
lock_out *unlock_1(lock_in *, CLIENT *cl);
```

[図 6]

```
typedef struct host_name {  
    char name[MAXNAME];  
} host_name;  
  
.void append_camera_list_1(host_name *, CLIENT *cl);  
void delete_camera_list_1(host_name *, CLIENT *cl);
```

〔图7〕



【図 3】

```

typedef struct camera_in{
    double tilt_angle;
    double pan_angle;
    int zoom;
} camera_in;

typedef struct func_out {
    int ret;
} func_out;

CLIENT *camera_open(char *host);
void camera_close(CLIENT *cl);

func_out *camera_pan_pos_1(camera_in *, CLIENT *cl);
func_out *camera_tilt_pos_1(camera_in *, CLIENT *cl);
func_out *camera_zoom_pos_1(camera_in *, CLIENT *cl);

```

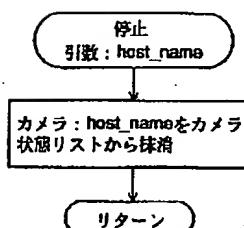
【図 5】

ホスト名	使用ホスト	位置	方位
host1	-----	(10, 15, 20)	(20, 35)
host2	host3	(45, 32, 20)	(0, 12)
host3	-----		
host4	host1		
⋮	⋮	⋮	⋮

← 80

【図 10】

【図 8】



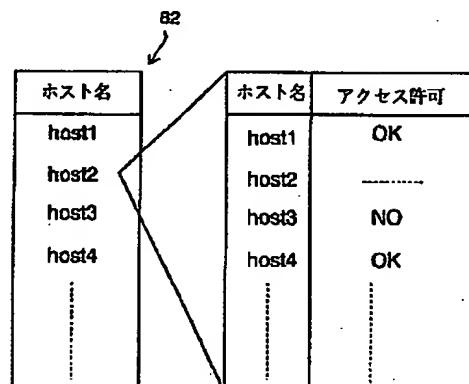
【図 9】

```

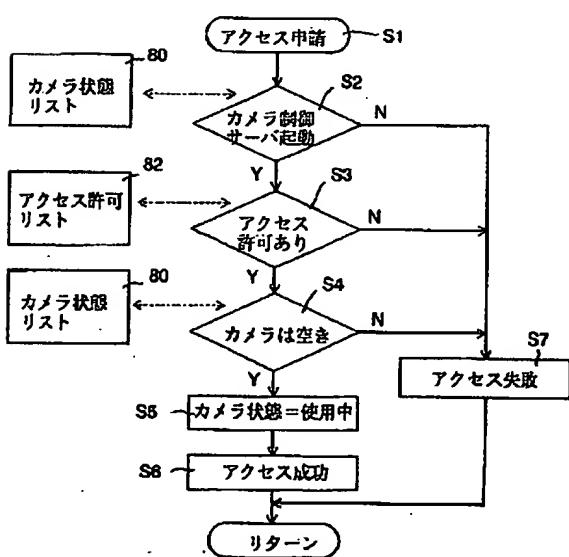
typedef struct access_in{
    char target_name[MAXNAME];
    char user_name[MAXNAME];
} access_in;

func_out *access_begin_1(access_in *, CLIENT *cl);
func_out *access_end_1(access_in *, CLIENT *cl);

```



【図 11】



【図 12】

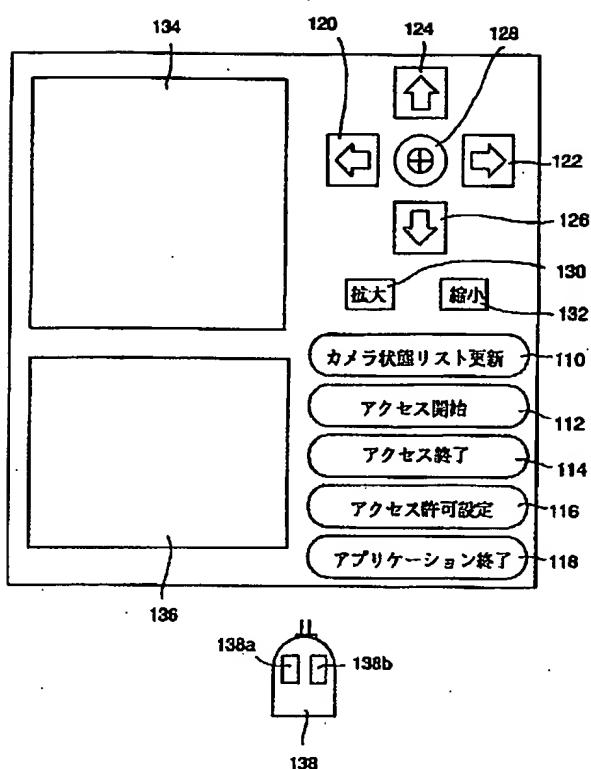
```

typedef struct change_in {
    char target_name[MAXNAME];
    int access_mode;
} change_in;

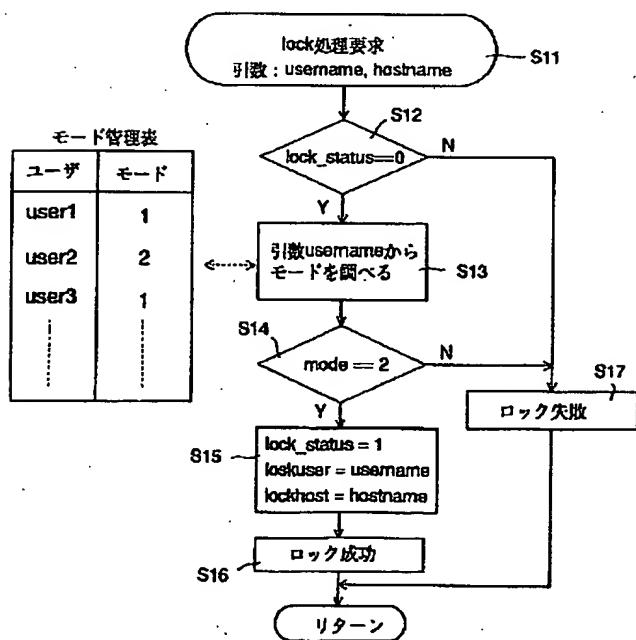
func_out *change_access_mode_1(change_in *, CLIENT *cl);

```

【図 13】



【図 15】



【図 14】

(a) モード管理表	
ユーザ	モード
user1	1
user2	2
user3	1
...	...

(b) 制限範囲管理表			
モード	操作	上限	下限
3	パン	5.0	-5.0
3	チルト	7.0	-7.0
3	ズーム	60	70

【図 16】

